



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC3745 — Testing
2022 - 2

Tarea 3

Fecha de entrega: Miércoles 14 de Septiembre del 2022 a las 23:59

Información General

La siguiente tarea contempla como objetivo principal entender la lógica de una técnica de análisis dinámico usada en la actualidad: code instrumentation. Al implementar un instrumentador sencillo el estudiante podrá ver las ventajas, el alcance y las desventajas de la técnica en general. No solo cuando uno implementa un instrumentador sino cuando uno usa alguno existente.

Objetivos

- Entender en detalle como funciona la técnica code instrumentation.
- Entender las ventajas, desventajas, y limitaciones de la técnica.
- Ser capaz de adaptar un instrumentador para detectar ejecuciones anormales en un programa analizado.

Contexto

En clase se desarrolló un **Instrumentation Profiler**, para contar cuantas veces se ejecutó una función. En la tarea 3 se pide modificar el profiler para detectar las funciones que reciben el mismo valor como argumento. La hipótesis detrás es que si una función recibe siempre el mismo valor como argumentos se puede implementar un cache para optimizar su rendimiento. Asumiendo claro, que la función no utilice variables globales o tenga alguna particularidad que provoque un devuelva diferentes resultados a un mismo argumento.

Por ejemplo:

```
def factorial(n):
    resultado = 1
    while n > 1:
        resultado *= n
        n -= 1
    return resultado

def abs(n):
    ...

print(factorial(3))
print(factorial(3))
print(factorial(3))
abs(3)
abs(2)
```

Figura 1: Ejemplo

Para el programa anterior, factorial siempre recibe el mismo argumento y puede ser optimizada. La función abs recibe distintos valores por lo que no puede ser optimizada.

Tarea

Desarrolle la siguiente actividad:

- Modificar el profiler para que imprima el nombre de las funciones que reciben el mismo valor como argumento siempre. (6 puntos).

Supuestos

- El programa a analizar está todo contenido en un solo archivo como en el ejemplo.
- El programa a analizar tiene varias funciones que reciben un único argumento.
- Las funciones solo reciben enteros positivos de argumentos.
- El programa siempre tendrá una lista de funciones al inicio y un código a ejecutar al final, como en el ejemplo anterior.

Entregables

- Usted debe diseñar y codificar al menos 3 ejemplos donde pueda mostrar que su profiler funciona correctamente.
- Debe subir el código de su tarea.
- Se debe subir la url a un vídeo corto de **máximo 2 minutos** donde explique la lógica de su solución y muestre sus ejemplos.

Archivos iniciales

En canvas se encuentra el código desarrollado por el profesor en clase con los siguientes archivos:

- *input code (folder)* – que contiene el código a ejecutar para probar el profiler
- *transformed-code (folder)* – que contiene el código con las instrucciones inyectadas al código de input, en la practica no se ejecuta este archivo solo sirve para ver si el instrumentador, modifiko el programa input correctamente.
- *instrumentaro.py* – código que recorre el AST e inyecta el código al inicio de cada función.
- *profiler.py* – lee el código de input, lo instrumenta, y ejecuta el código instrumentado, finalmente imprime las métricas colectadas en consola.

Para la tarea usted debe modificar o agregar código a los archivos anteriormente mencionados según corresponda.

Reportar problemas en el equipo

En el caso de que algún integrante no aportara como fue esperado en la tarea, podrán reportarlo enviando un correo con asunto **Problema Equipo {NumeroGrupo} Testing** a juanandresarriagada@uc.cl con copia a juanpablo.sandoval@uc.cl explicando en detalle lo ocurrido. Posterior a eso se revisara el caso en detalle con los involucrados y se analizara si corresponde aplicar algún descuento. Instamos a todas las parejas que mantengan una buena comunicación y sean responsables con el resto de su equipo para evitar problemas de este estilo.

Restricciones y alcances

- Su programa debe ser desarrollado en **Python 3.9 o superior**.
- Los archivos de código entregados deben terminar con la extensión **.py**.
- En caso de dudas con respecto al enunciado deben realizarlas en un foro relacionado a la tarea que se encontrara disponible en canvas.
- Si no se encuentra especificado en el enunciado, supón que el uso de cualquier librería de Python adicional a las utilizadas en el código base se encuentran prohibidas. En caso de que estimes necesario podrás preguntar en el foro de la tarea por el uso de alguna librería adicional.

Entrega

Código: Deberán entregar el código que utilizaron para su tarea junto con los ejemplos que utilizaron para probar su funcionamiento. La entrega se realizara por medio de un buzón de canvas habilitado para esta tarea.

Vídeos: Deben subir el vídeo a YouTube e incluir la url en un archivo .txt que entregaran en el buzón junto con el código. El vídeos deben estar en configuración **"No Listado"**, o sea, **que solo se puedan acceder desde un url**¹.

Atraso: Se efectuara un descuento por entregar tareas atrasadas. Se descontara 0.5 si la tarea se entrega con menos de una hora de retraso. El puntaje final en caso de atraso seria calculado mediante la siguiente formula:

$$PuntajeFinal = PuntajeObtenido - (0,5 + 0,05 \cdot k)$$

, donde k es el numero de horas de retraso menos uno.

Integridad académica

Este curso se adscribe al Código de Honor establecido por la Escuela de Ingeniería. Todo trabajo evaluado en este curso debe ser hecho **individualmente** o en **los grupos asignados** según sea definido en la evaluación y **sin apoyo de terceros**. Se espera que los alumnos mantengan altos estándares de honestidad académica, acorde al Código de Honor de la Universidad. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Pregrado de la Escuela de Ingeniería.

¹Con el fin de poder resguardar la integridad académica

¡Éxito con la Tarea! :D